

Ferramentas para desenvolvimento CASE

Msc. Lílian Simão Oliveira

Maio de 2015



Agenda

1. Introdução
2. O que é Ferramenta Case?
3. Blocos Construtivos de Case
4. Taxonomia de Ferramentas Case
5. Ambientes Case Integrados
6. Arquitetura de Integração
7. Repositório Case
8. Vantagens e Problemas
9. Referência Bibliográfica



1. Introdução



Fonte: Getty Images

- Uma oficina precisa ter:
 - Uma coleção de ferramentas úteis
 - Uma disposição organizada
 - Um hábil artesão



Fonte: Getty Images

2. O que é uma Ferramenta Case?



Fonte: Getty Images

CASE – *Computer-Aided Software Engineering* -
Engenharia de Software Apoiada por Computador

ferramentas automatizadas
que auxiliam gerentes e
profissionais de engenharia
de software em toda
atividade com o processo de
software.

Pressman, 2002

Outros conceitos

"CASE designa um conjunto de ferramentas que **auxiliam** um programador ou um gestor de projetos durante uma ou mais **fases do processo de desenvolvimento** de software, incluindo a manutenção".

Silva(2005) apud B. Terry (1990)

"CASE *é a* utilização de meios de **suporte** baseados em computador no **processo de desenvolvimento** de software".

Software Engineering Institute



Por que é importante?

- Diminuição de esforço
- Novos pontos de vista
- Decisões elaboradas
- Melhora na qualidade



Qual é o produto do trabalho?

- Produtos com qualidade
- Trabalhos adicionais

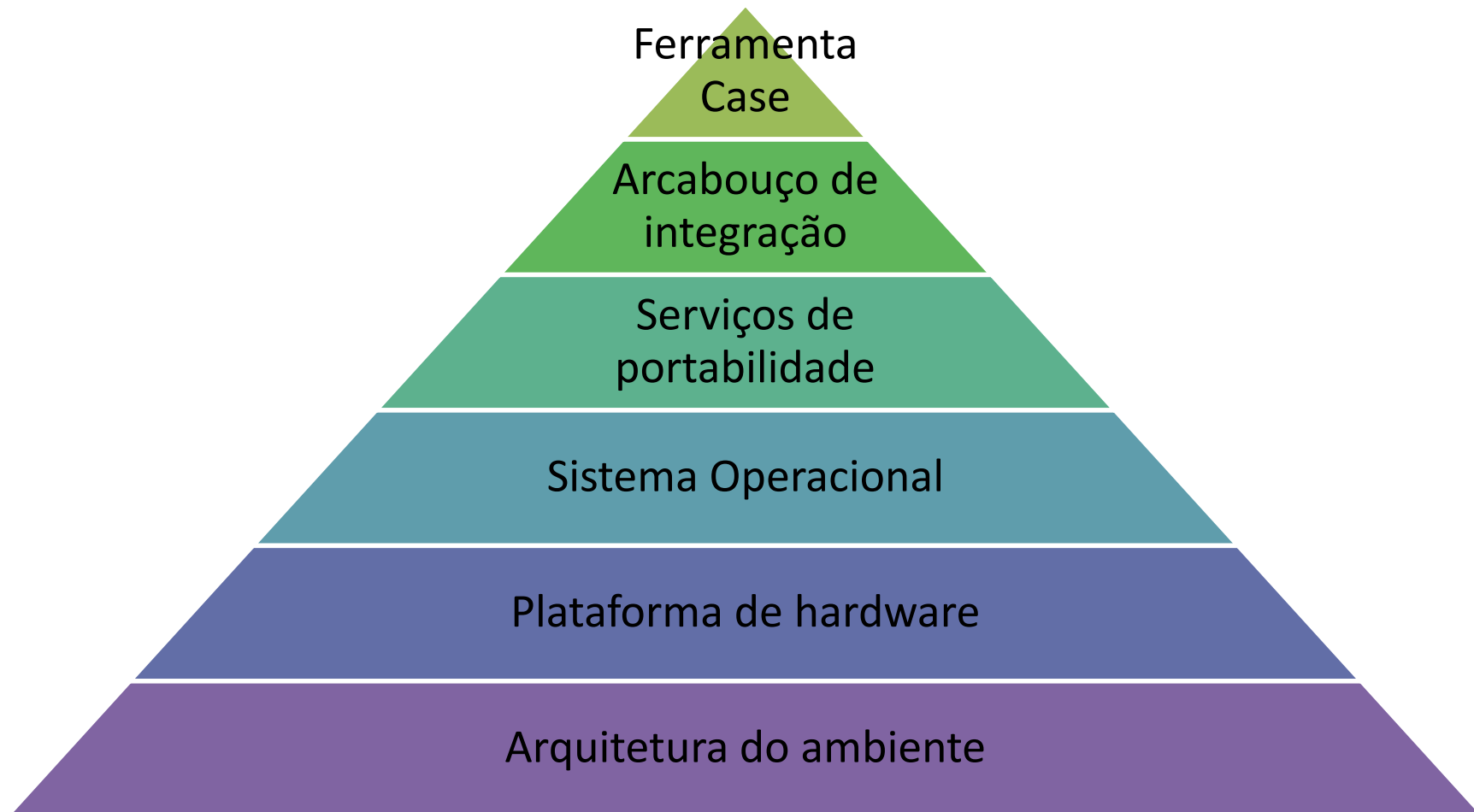


Como garantir uso correto?

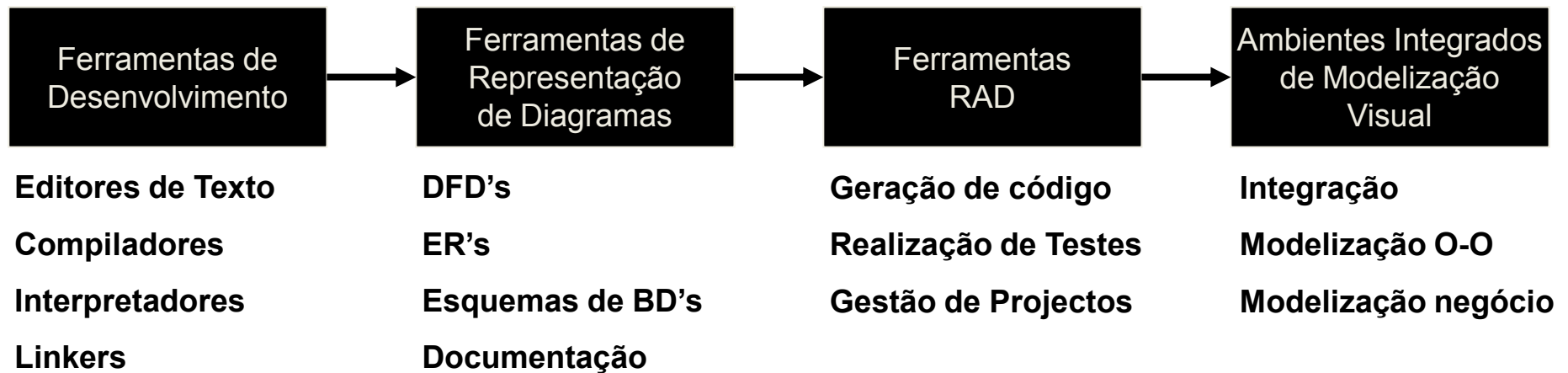
- Complementar prática confiáveis
- Estabelecer um arcabouço de processo de software



3. Blocos construtivos

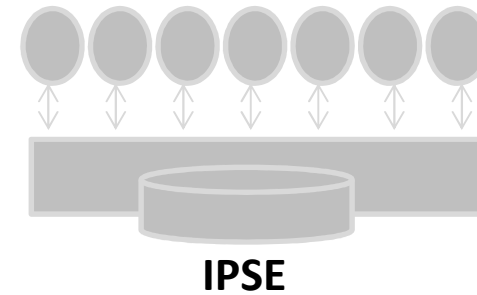


Evolução das Ferramentas de apoio ao desenvolvimento de software

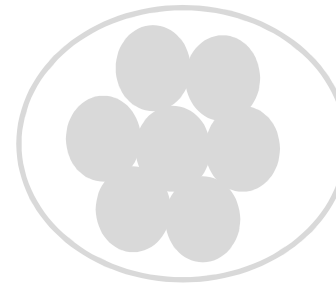


Opções de integração

IPSE

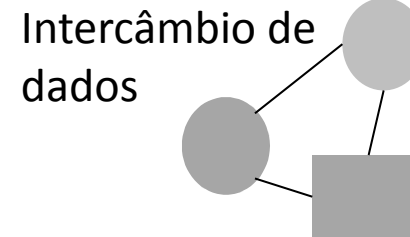


Fonte única



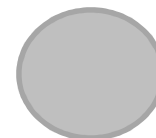
Fonte única

Fonte e associação de ferramentas



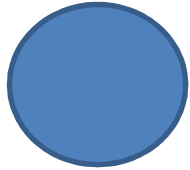
Fontes e associações de ferramentas

Solução pontual

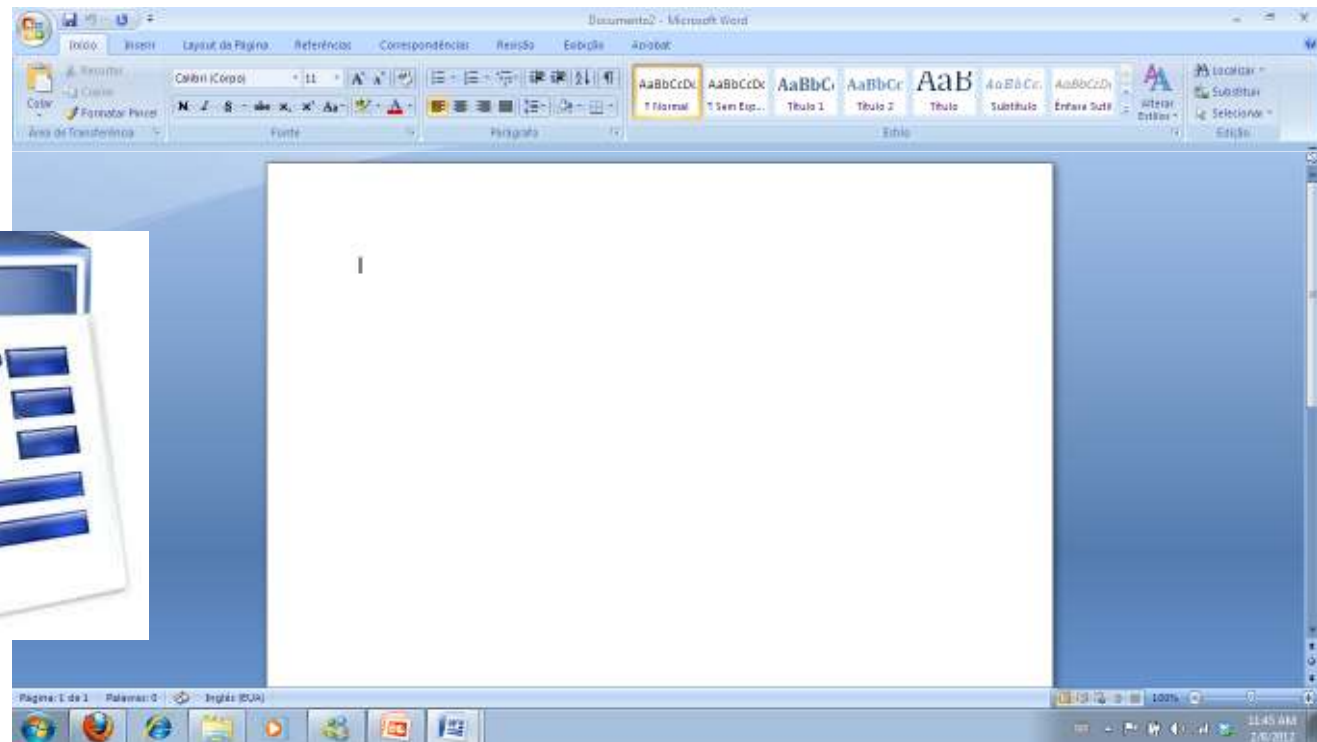


Ferramenta individual
(Solução Pontual)

Exemplo de Ferramenta Individual

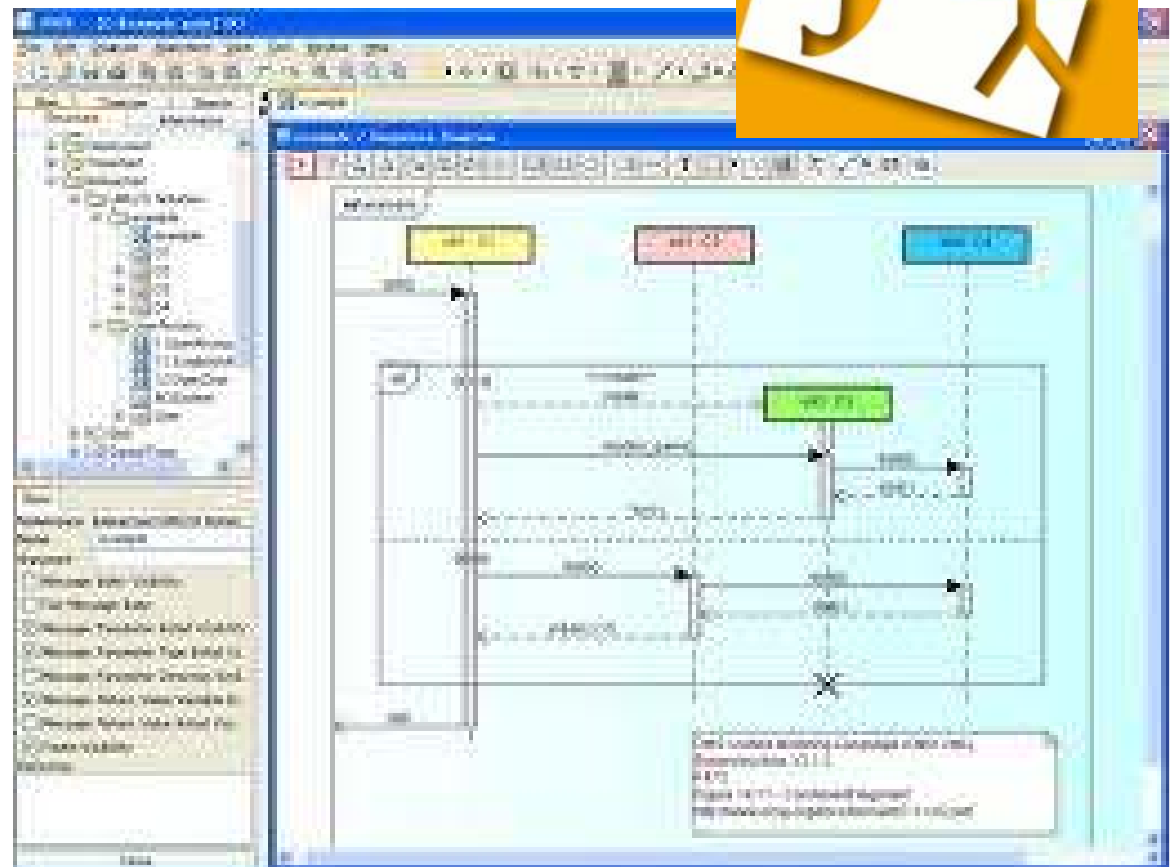
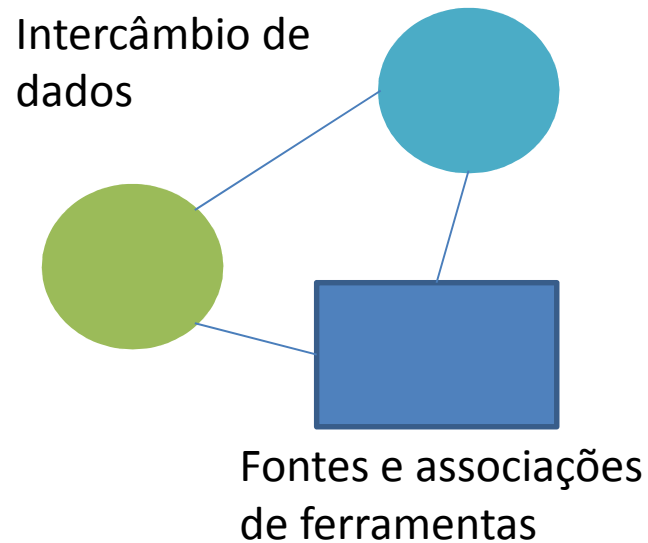


Ferramenta individual
(Solução Pontual)



Fonte: Google Images

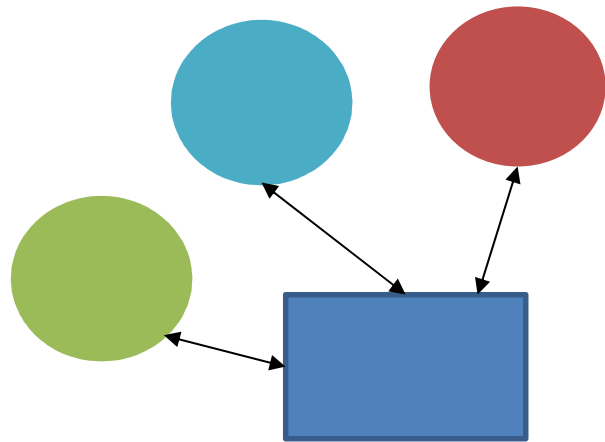
Exemplo de associações de ferramentas



Fonte: Google Images

Exemplo de associações de ferramentas

Consórcios e padrões

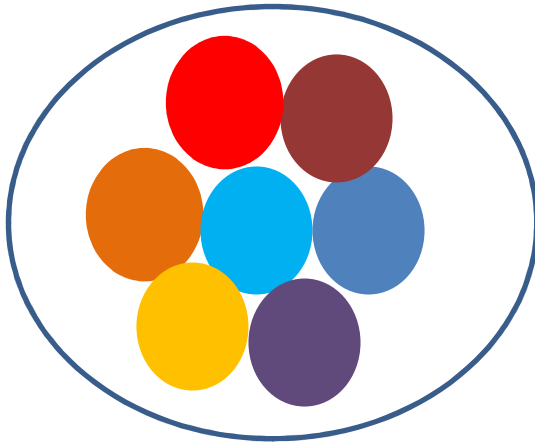


Mesma base de dados

Fonte: Google Images

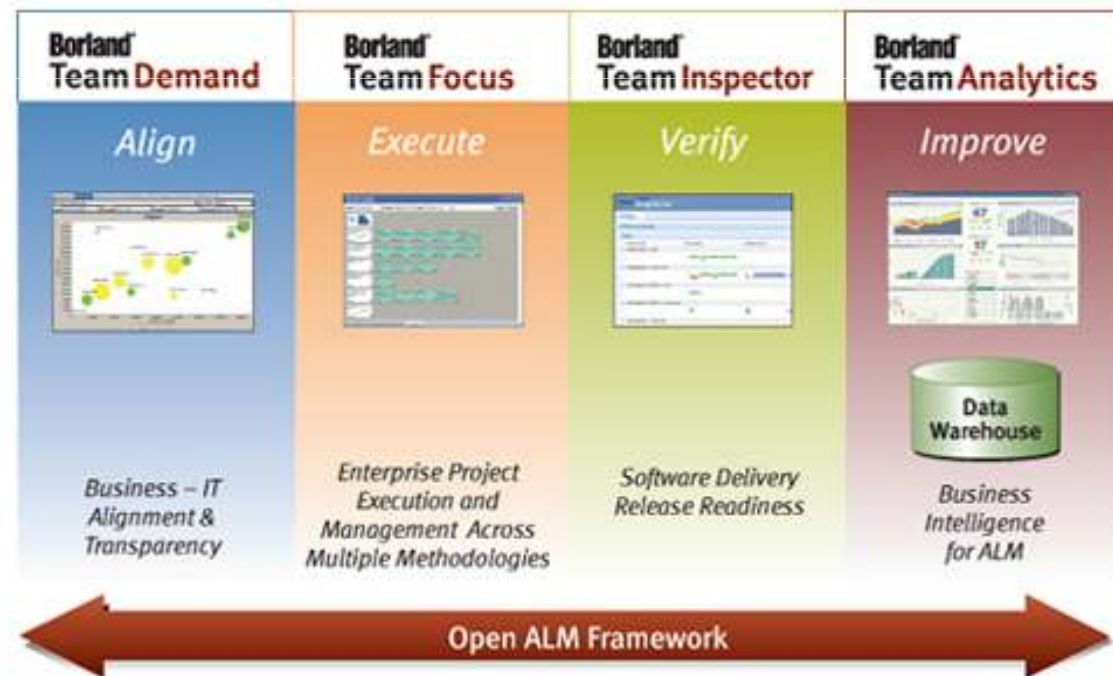


Exemplo de Fonte única



Fonte única

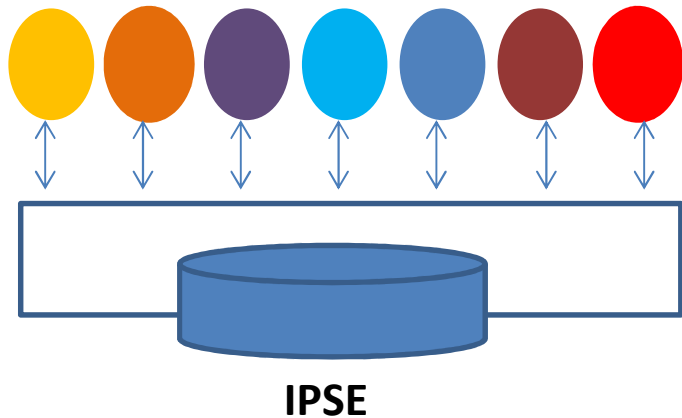
Borland®
THE OPEN ALM COMPANY



Fonte: Borland -
<http://www.borland.com/>

Exemplo de IPSE

(Integrated Project Support Environment)



Fonte: Site IBM



Software IBM > Produtos >

Produtos por capacidade

Buscador de produtos de Software IBM

Análítica Avançada	<ul style="list-style-type: none">Experiência da Web de funcionáriosExperiência do cliente na Web
Aplicativos Analíticos	<ul style="list-style-type: none">Aplicativos para Análise da Cadeia de SuprimentosGerenciamento de sistemas de mainframe
Colaboração Social	<ul style="list-style-type: none">Acesso remoto e corporativoColaboração em tempo realE-mailProdutividade e conjuntos de escritório
Comunicações Unificadas	<ul style="list-style-type: none">Comunicações Unificadas
Conectividade e Integração	<ul style="list-style-type: none">Base de integraçãoIntegração B2BIntegração com a nuvemSistema de mensagensTransferência de arquivo gerenciada
Controle de Informações	<ul style="list-style-type: none">Gerenciamento de ciclo de vida de dadosGerenciamento de dados de qualidade

IBM Software

- Produtos
 - Produtos por necessidade de negócio
 - Produtos por Categoria
 - Produtos do A ao Z
 - Produtos por palavra chave
- Serviços
- Downloads (US)
- Como comprar
- Treinamentos
- Subscrição e Suporte
- Suporte (US)

Links relacionados

- Demonstrações de Software IBM
- Biblioteca de recursos técnicos de software (US)
- Portal de Tecnologia IBM developerWorks

4. Taxonomia de ferramentas CASE

Critérios:

- Por função
- Por seu papel como instrumentos para gerentes ou pessoal técnico
- Por uso em uma fase do processo
- Pela arquitetura (HW e SW)
- Por sua origem ou custo

Fonte: Getty Images



Critério: Fase do processo

- Ferramentas **Upper-Case** são aplicações que se especializaram na fase de concepção do software (ferramentas de análise e especificação e/ou modelação de requisitos).
- Ferramentas **Lower-Case** são aplicações utilizadas na fase de implementação (ferramentas de desenho técnico, de edição e compilação de código e de testes).



Fase do processo - Detalhada

- Modelação de processos de negócio
- Modelação de análise e projeto do sistema
- Desenho de bases de dados
- Programação de aplicações
- Gestão de alterações no software
- Testes
- Orientadas para a Gestão de Projetos



Modelagem de processos de negócio

- verificar como os objetivos estratégicos de negócio são concretizados em processos
- notações e diagramas para a representação de informação do negócio (cadeia de valor, responsabilidades e funções da organização), técnicas de simulação e análise de custos (por exemplo, análise ABC)
- Exemplos: Aris Toolset (www.idsscheer.com), Mega Suite (www.mega.com), Provision (www.proformacorp.com).



Modelagem de análise e projeto do sistema

- modelos de processos com os modelos e requisitos a implementar
- Exemplos:
- OO → o Rose (www.rational.com), o Paradigm Plus (www.cai.com), o GDPro (www.advancedsw.com).
- Estruturada → System Architect (www.popkin.com), o PowerDesigner (www.sybase.com) e o Silverrun (www.silverrun.com).



Desenho de Base de Dados

- aparecem na seqüência das ferramentas anteriores (muitas vezes de forma integrada), mas especializaram-se na **definição lógica e física** da estrutura das **bases de dados**.
- Exemplos: System Architect (www.popkin.com) o PowerDesigner (www.sybase.com) e o Erwin (www.cai.com).



Programação de aplicação

- ambiente único e integrado
- Possui:
 - Edição de programas,
 - Concepção da interface
 - Interpretadores
 - Compiladores
 - Geradores de código
 - *Debuggers*
- Exemplos: Visual Basic e Visual C++ (www.microsoft.com), Delphi (www.borland.com) e Powerbuilder (www.sybase.com).



Gestão de alteração no software

- suportam o trabalho em equipe
- Funcionalidades:
 - gestão de versões,
 - mecanismos de *check-in* e *check-out*
 - gestão da configuração e distribuição do software.
- Exemplos: Visual Sourcesafe (www.microsoft.com) e ClearQuest (www.rational.com).



Teste

- Permitem:
 - Criação de regras de testes
 - Geração de *scripts* de testes
 - Definição de dados
 - Controle e a gestão de erros
 - Obtenção de estatísticas
- Exemplos: Suite TestStudio (www.rational.com) e TestWorks (www.soft.com), Junit .



Orientadas para gestão de projetos

- Funcionalidades:
 - planejamento e estimativa de tempos, custos e recursos
 - utilização e medição de recursos do projeto
 - definição de responsabilidades
 - auxílio na aplicação de uma metodologia de desenvolvimento de software
- Exemplos: Project (www.microsoft.com) e Juggler (www.cse.dcu.ie/catalyst).



Categoria: Função

- Pressman (2002) subdivide nas seguintes categorias:
 - Ferramentas de engenharia de processo de negócio
 - Ferramentas de modelagem e gestão de processo
 - Ferramentas de planejamento de projeto
 - Ferramentas de análise de risco
 - Ferramentas de gestão de projeto
 - Ferramentas de rastreamento de requisitos
 - Ferramentas de métricas e gestão
 - Ferramentas de documentação



Categoria: Função (cont.)

- Pressman (2002) subdivide nas seguintes categorias:
 - Ferramentas de software básico
 - Ferramentas de garantia da qualidade
 - Ferramenta de gestão de base de dados
 - Ferramentas de gestão de configuração de software
 - Ferramentas de análise e projeto
 - Ferramentas PRO/SIM
 - Ferramentas de projeto e desenvolvimento de interfaces

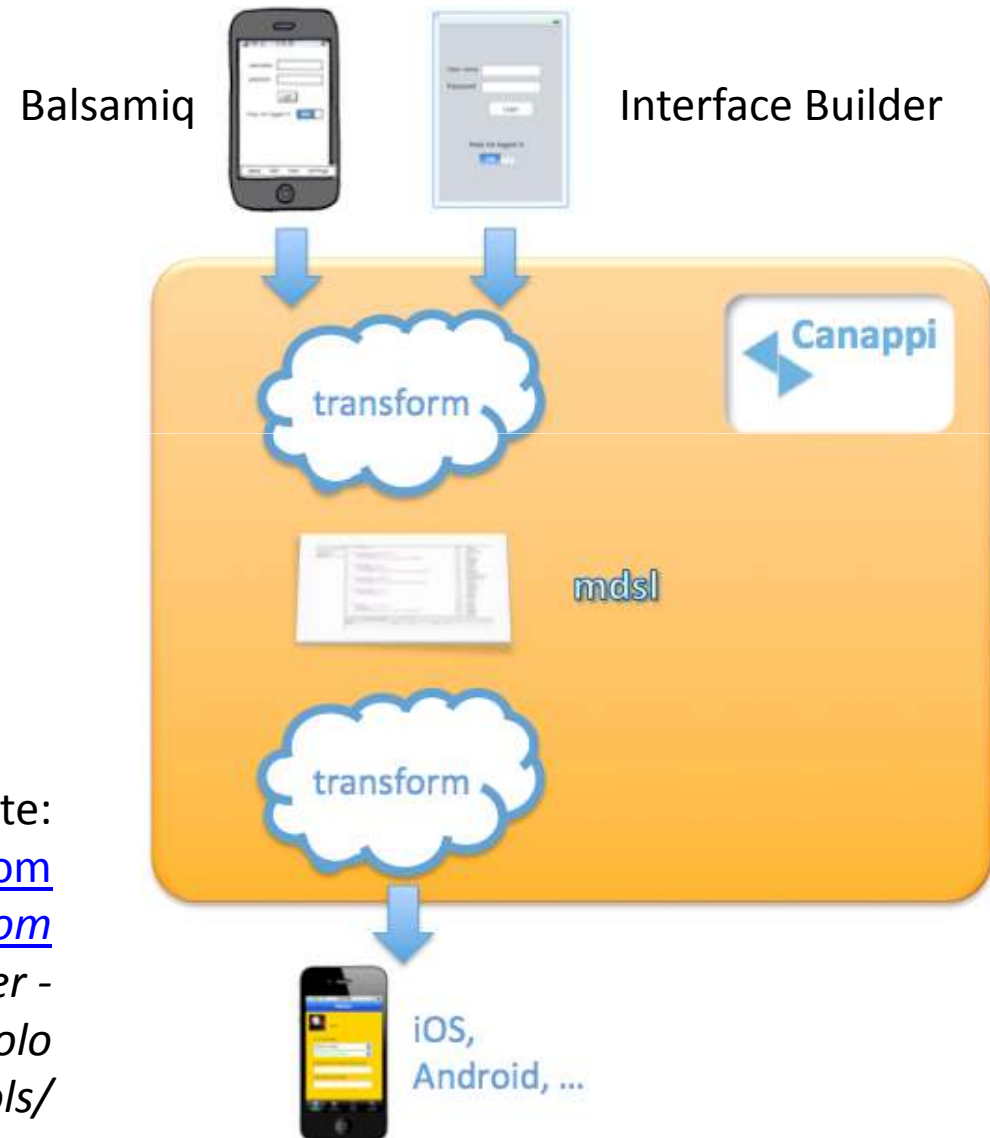


Categoria: Função (cont.)

- Pressman (2002) subdivide nas seguintes categorias:
 - Ferramentas de prototipação
 - Ferramentas de programação
 - Ferramentas de desenvolvimento Web
 - Ferramentas de integração e teste
 - Ferramentas de análise estatística
 - Ferramenta de análise dinâmica
 - Ferramentas de gestão de teste
 - Ferramentas de teste cliente/servidor
 - Ferramentas de reengenharia



Exemplo de ferramentas de prototipação



Fonte:

Canappi - <http://www.canappi.com>

Balsamiq - www.balsamiq.com

Interface Builder -
<https://developer.apple.com/technologies/tools/>

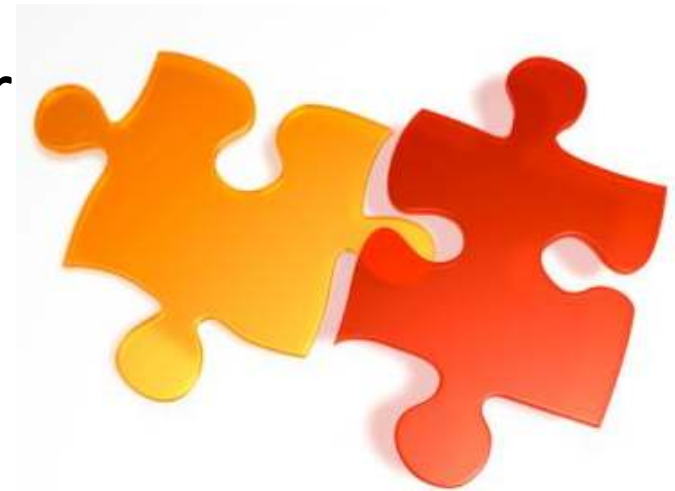
5. Ambientes Case Integrados

- *i-Case* – benefícios:
 - Transferência constante de informações
 - Redução do esforço necessário para realizar atividades guarda-chuva
 - Aumento do controle de projeto
 - Maior coordenação entre os membros da equipe

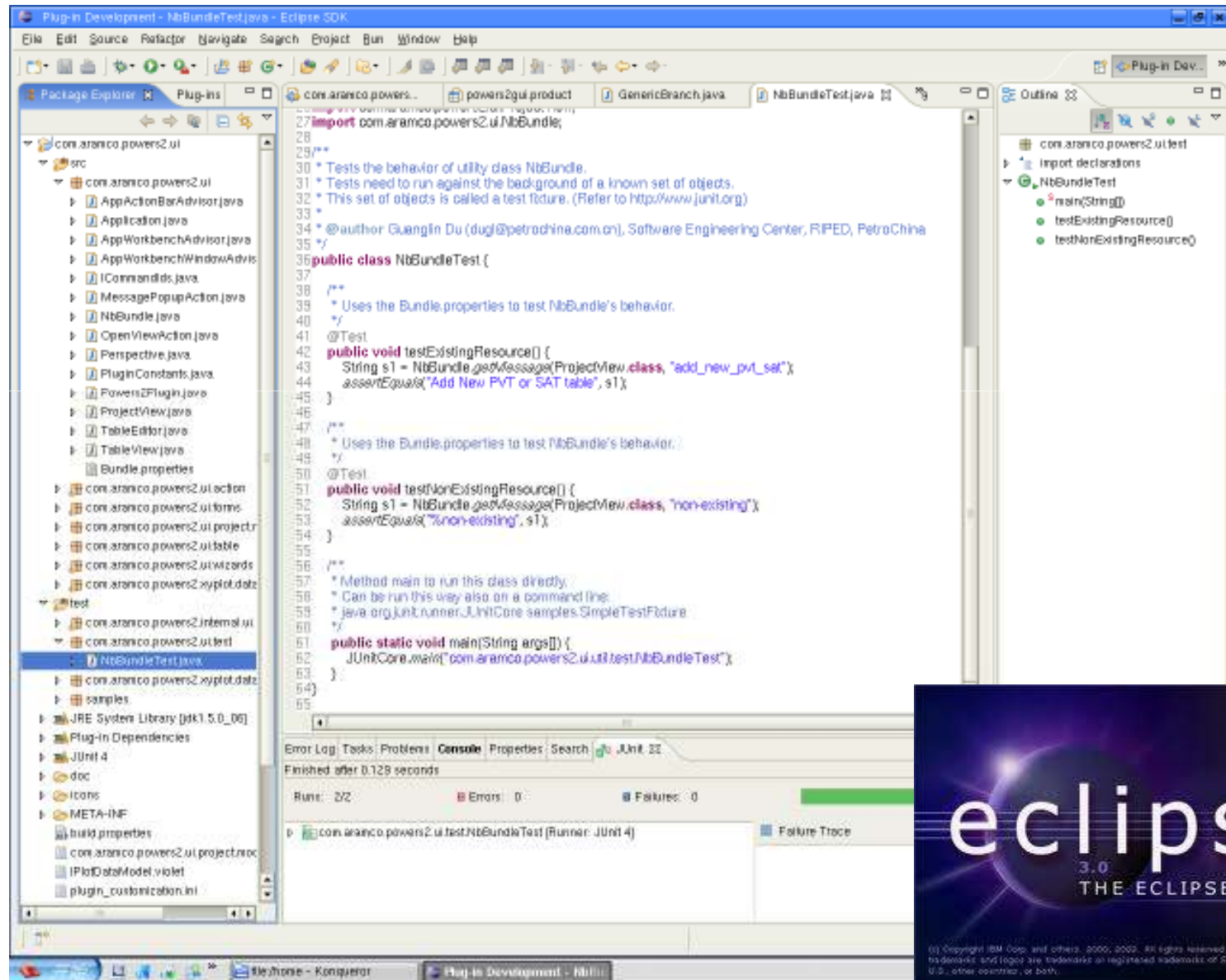


i-Case - Características

- Compartilhar informações
- Rastreamento de modificações
- Controle de Versão
- Acesso direto
- Suporte automatizado
- Interface homem/computador
- Coletar métricas



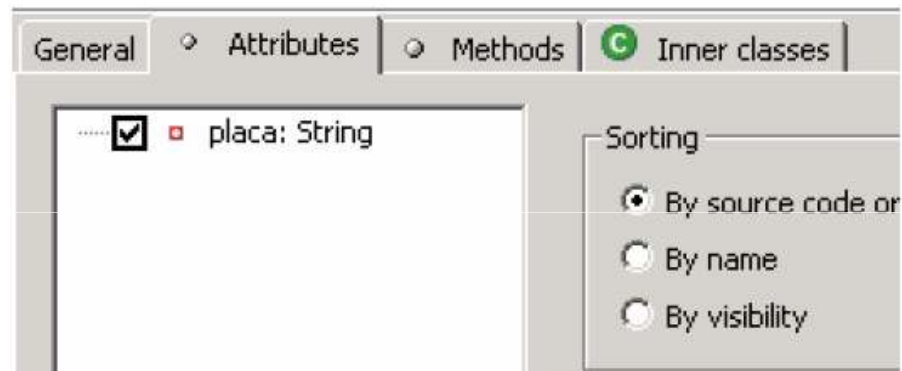
Exemplo: IDE Eclipse



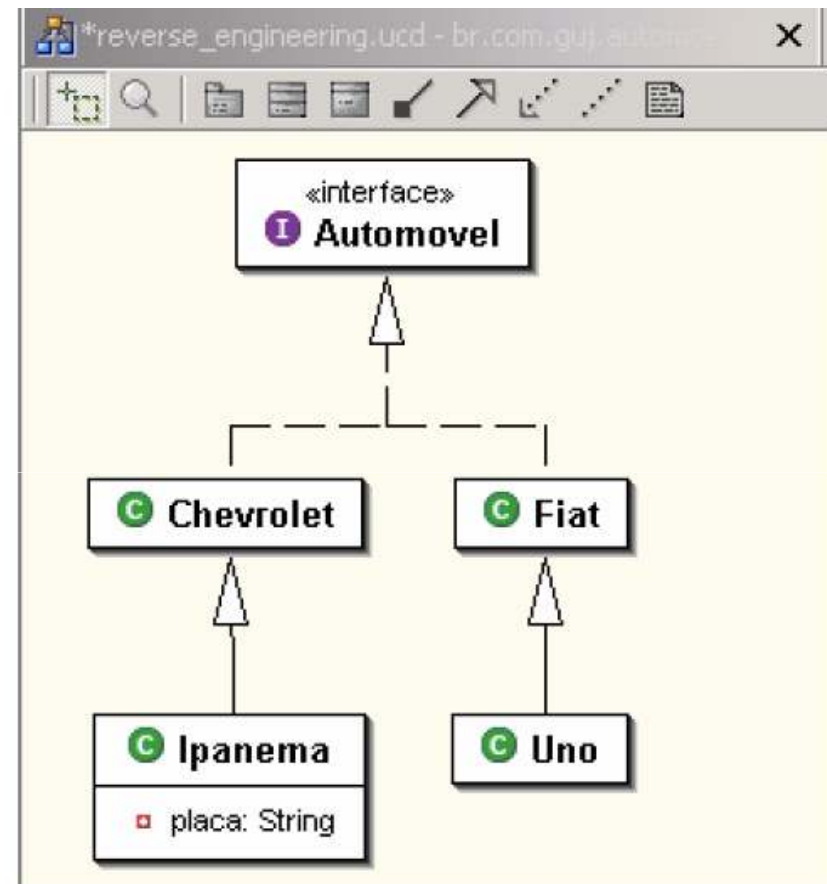
UML – Eclipse – Plugin Omondon

View selector

UML class view selector



9. View Selector



10. Diagrama de Classes

```
package br.com.guj.automovel;

public class Ipanema extends Chevrolet {
    private String placa;
}
```

6. Arquitetura de Integração

Camada de interface com o usuário
Caixa de ferramentas da interface
Protocolo de apresentação

Serviços de gestão de ferramentas

Ferra-
menta
CASE

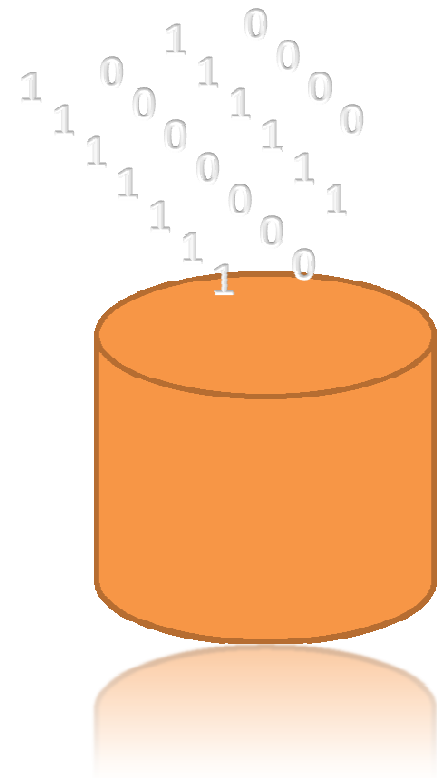
Camada de
ferramentas

Camada de gestão de objetos
Serviços de integração
Serviços de gestão de configuração

Camada de repositório compartilhado
Base de dados CASE
Funções de controle de acesso

7. Repositório CASE

- Papel do repositório em i-Case:
 - Integridade dos dados
 - Compartilhamento das informação
 - Integração dados/ferramenta
 - Integração dados/dado
 - Imposição de metodologia
 - Padronização de documentos



7. Repositório CASE

- O que é armazenado:
 - O problema a ser resolvido
 - Informação sobre o domínio do problema
 - Solução do sistema
 - Regras e instruções do processo
 - Plano do projeto, recursos e histórico
 - Informação do contexto organizacional



7. Repositório CASE

Base de Dados Comercial

- Características:
 - Armazenagem de dados não redundante
 - Acesso de alto nível
 - Independência dos dados
 - Controle de transações
 - Segurança
 - Consultas de dados e relatórios ad hoc
 - Abertura
 - Suporte a multiusuário



7. Repositório CASE

Características particulares

- Armazenagem de estruturas de dados
- Imposição de integridade
- Ferramenta com interface rica em semântica
- Gestão de processos/projeto



7. Repositório CASE

- Características:
 - Gerência de versões
 - Rastreamento de dependência e gestão de modificações
 - Rastreamento de requisitos
 - Gestão configuração
 - Pistas de auditoria



8. Vantagens e Problemas

- Vantagens:
 - Uniformização
 - Reutilização
 - Automatização
 - Diminuição do tempo de desenvolvimento
 - Integração
 - Demonstração da consistência
 - Qualidade



8. Vantagens e Problemas

- Desvantagens:
 - Tempo de aprendizado
 - Mapear o processo de negócio em requisitos
 - Integração entre desenho lógico e estrutura física



9. Referência Bibliográfica

- Róger S. Pressman – Engenharia de Software – 5a edição – Mc Graw Hill - 2002
- Silva, Alberto Manuel Rodrigues & VIDEIRA, Carlos Alberto Escaleira, “UML, Metodologias e Ferramentas CASE”, Edições Centro Atlântico, 2001.

